Instituto Federal do Sul de Minas Gerais

Projeto e Análise de Algoritmos

Aula 07 – Notações θ, Ω, ω, o

humberto@bcc.unifal-mg.edu.br
douglas@bcc.unifal-mg.edu.br

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL DE MINAS GERAIS

Última aula

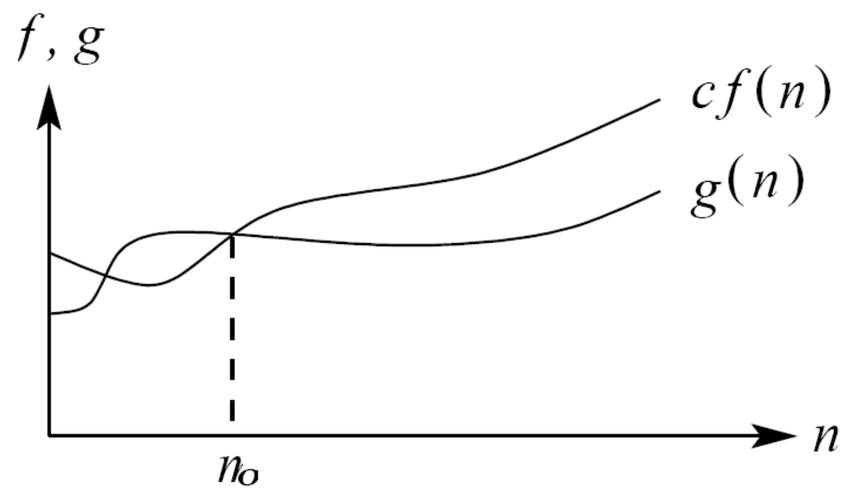
Notação O

- Uma função $f(n)$ domina assintoticamente outra função $g(n)$ se existem duas constantes positivas
  - c e n_o

Última aula

Notação O

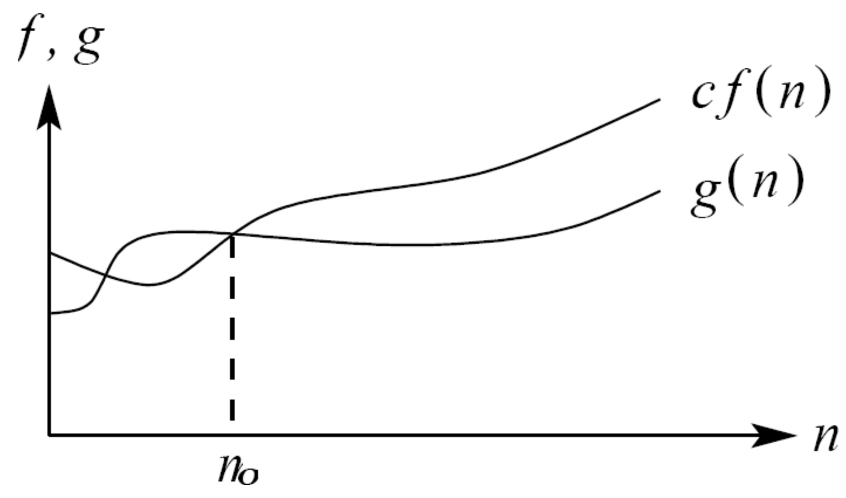
- Uma função $f(n)$ domina assintoticamente outra função $g(n)$ se existem duas constantes positivas
  - c e n_o
- tais que, para qualquer
  - $n >= n_o$,



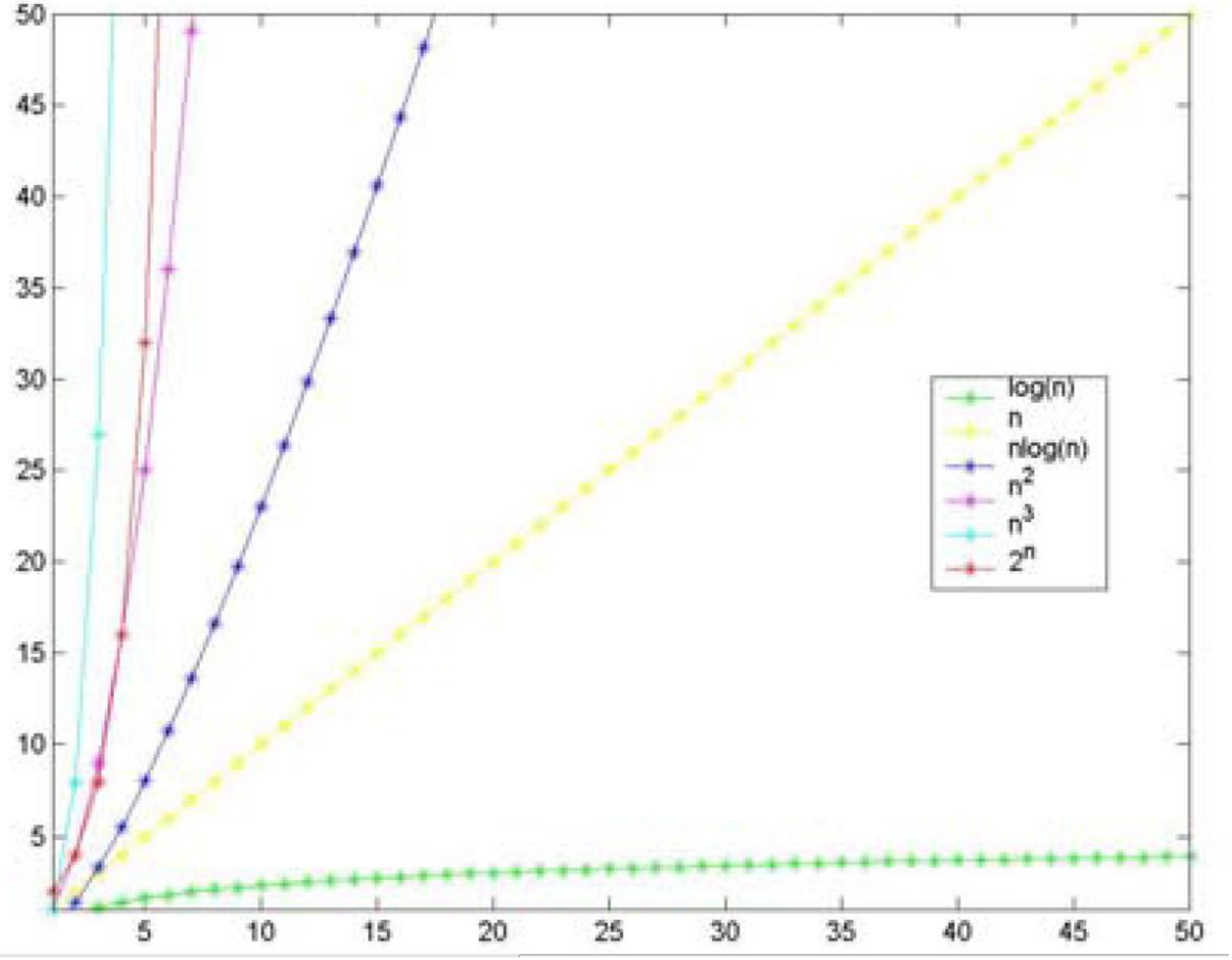
Última aula

Notação O

- Uma função $f(n)$ domina assintoticamente outra função $g(n)$ se existem duas constantes positivas
  - c e n_o
- tais que, para qualquer
  - $n >= n_o$,
- temos
  - $g(n) <= c \cdot f(n)$



Comparativo..



Outras notações

- Assim como a notação *O* fornece uma maneira assintótica de dizer que uma função é “menor ou igual a”outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;

Outras notações

- Assim como a notação *O* fornece uma maneira assintótica de dizer que uma função é “menor ou igual a”outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;
  - Θ

Outras notações

- Assim como a notação *O* fornece uma maneira assintótica de dizer que uma função é "menor ou igual a"outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;

    - Θ
    - Ω

Outras notações

- Assim como a notação *O* fornece uma maneira assintótica de dizer que uma função é "menor ou igual a"outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;
  - Θ
  - Ω
  - ω

Outras notações

- Assim como a notação O fornece uma maneira assintótica de dizer que uma função é “menor ou igual a”outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;

  - Θ
  - Ω
  - ω
  - o

Notação Ω

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - ‘O’ define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - ‘O’ define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

- Exemplos:

$$n^4 \in \Omega(n^3)$$

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - ‘O’ define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

- Exemplos:

$$n^4 \in \Omega(n^3)$$

$$n \in \Omega(1)$$

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - 'O' define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

- Exemplos:

$$n^4 \in \Omega(n^3)$$

$$n \in \Omega(1)$$

$$3 \cdot \log(n) \in \Omega(\log(n))$$

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - 'O' define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

- Exemplos:

$$n^4 \in \Omega(n^3)$$

$$n \in \Omega(1)$$

$$3 \cdot \log(n) \in \Omega(\log(n))$$

$$1 \in \Omega(1)$$

Notação Ω

- A notação Ω é bem parecida com a notação O;
  - ‘O’ define um limite assintótico superior, e;
  - Ω define um limite assintótico inferior.

- Exemplos:

$$n^4 \in \Omega(n^3)$$

$$n \in \Omega(1)$$

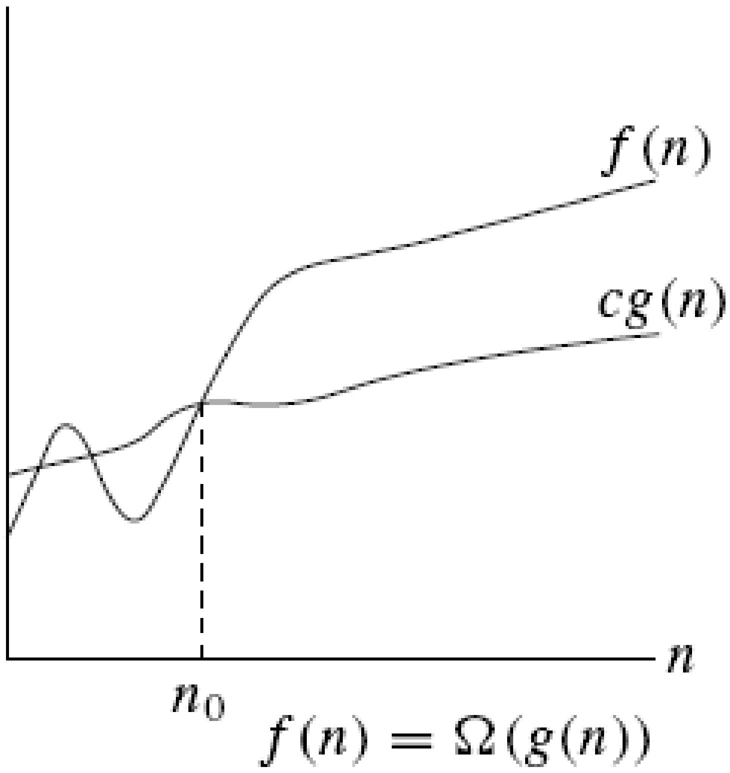
$$3 \cdot \log(n) \in \Omega(\log(n))$$

$$1 \in \Omega(1)$$

$$n! \in \Omega(2^n)$$

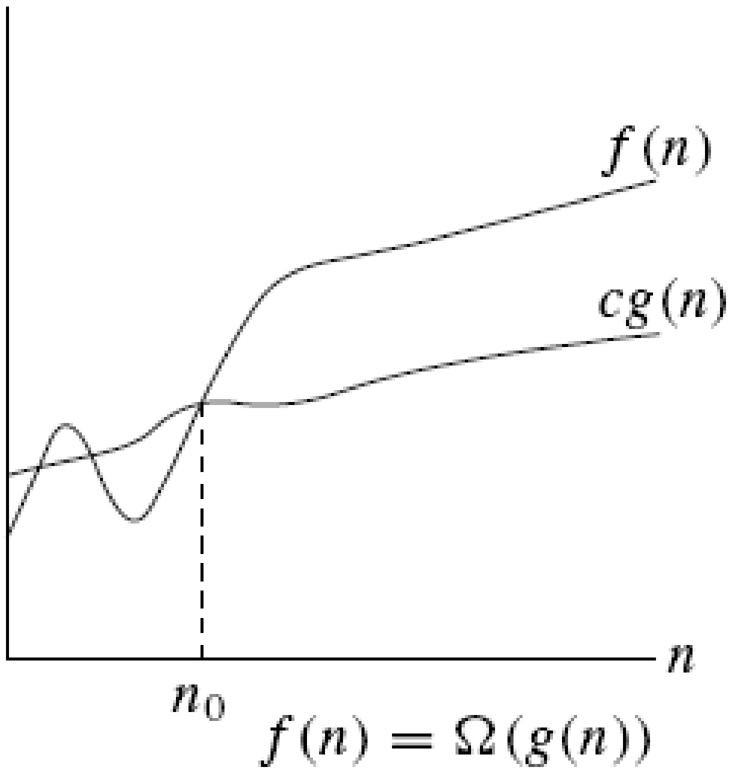
Notação Ω

- Limite assintótico inferior



Notação Ω

- Limite assintótico inferior



$$\Omega(g(n)) = \{f(n):\ \exists\ c\ e\ n_0\ > 0$$
$$\mid 0 \leq c \cdot g(n) \leq f(n) \quad \forall \quad n \geq n_0\}$$

Notação Ω

- Na prática a notação Ω não é vista sozinha em análises de algoritmos;

Notação Ω

- Na prática a notação Ω não é vista sozinha em análises de algoritmos;
  - Pelo motivo de não interessar para a análise de algoritmos;

Notação Ω

- Na prática a notação Ω não é vista sozinha em análises de algoritmos;
  - Pelo motivo de não interessar para a análise de algoritmos;
  - A notação O possui sua importância, pois o programador conclui que seu algoritmo é no máximo tão complexo a uma função.

Notação Ω

- Na prática a notação Ω não é vista sozinha em análises de algoritmos;
  - Pelo motivo de não interessar para a análise de algoritmos;
  - A notação O possui sua importância, pois o programador conclui que seu algoritmo é no máximo tão complexo a uma função.
  - Mas no mínimo tão complexo, como a notação Ω descreve, não é importante para conclusões práticas sobre algoritmos.

Notação Ω

- Na prática a notação Ω não é vista sozinha em análises de algoritmos;
  - Pelo motivo de não interessar para a análise de algoritmos;
  - A notação O possui sua importância, pois o programador conclui que seu algoritmo é no máximo tão complexo a uma função.
  - Mas no mínimo tão complexo, como a notação Ω descreve, não é importante para conclusões práticas sobre algoritmos.
- Ω vem na maioria das vezes acompanhada a notação Θ;
  - Como um complemento na análise, nunca sozinha...

Notação θ

Notação θ

- Conhecida também como “limite firme” ou “limite assintoticamente restrito”.

Notação θ

- Conhecida também como "limite firme" ou "limite assintoticamente restrito".

- A notação O, apesar de fornecer informações sobre a complexidade do algoritmo, nem sempre nos revela algo importante;

Notação θ

- Conhecida também como **“limite firme” ou “limite assintoticamente restrito”.**

- **A notação O**, apesar de fornecer informações sobre a complexidade do algoritmo, **nem sempre nos revela algo importante**;

- Não faz sentido, para algum algoritmo, dizer que suas complexidade é por exemplo $O(n!)$.
  - Ou faz?

Notação θ

- Conhecida também como "limite firme" ou "limite assintoticamente restrito".

- **A notação O**, apesar de fornecer informações sobre a complexidade do algoritmo, **nem sempre nos revela algo importante**;

- Não faz sentido, para algum algoritmo, dizer que suas complexidade é por exemplo *O(n!)*. Ou faz?

- Exemplos da falta de precisão de O:

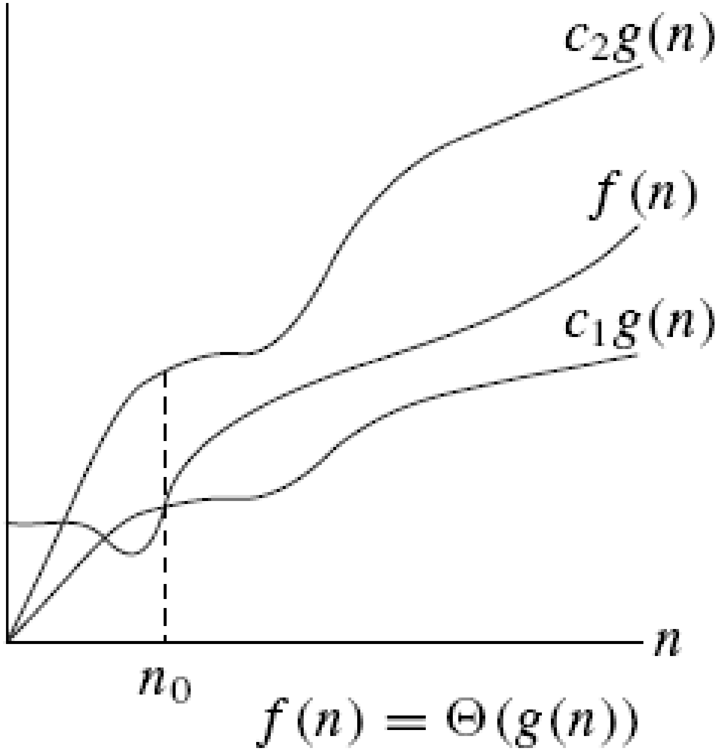
$$n \in O(n^3)$$
$$n \in O(n^4)$$
$$n \in O(n^5)$$
$$n \in O(n^{1000})$$
$$n \in O(2^n)$$
$$n \in O(n!)$$

Notação θ

- Uma função $f(n)$ pertence ao conjunto $\theta(g(n))$ se existem constantes positivas n_0, c_1 e c_2

Notação θ

- Uma função *f(n)* pertence ao conjunto *θ(g(n))* se existem constantes positivas n_0, c_1 e c_2 tais que ela possa ser "imprensada" entre
  c1.g(n) e *c2.g(n)*,
  para um valor de *n* suficientemente grande.



$$\Theta(g(n)) = \{f(n):\ \exists\ c_1, c_2\ e\ n_0\ > 0$$
$$|\ 0 \le c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n)\quad \forall\quad n \ge n_0\}$$

Notação θ

$$\Theta(g(n)) = \{f(n):\ \exists\ c_1, c_2\ e\ n_0\ > 0 \\ \mid 0 \le c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n)\quad \forall\quad n \ge n_0\}$$

- Exemplo: $$\frac{n^2}{2} - 3n \in \Theta(n^2)$$

- Para isso, devemos definir constantes c_1, c_2 e n_0 tais que:

$$c_1 n^2 \le \frac{1}{2} n^2 - 3n \le c_2 n^2$$

- Encontre constantes que satisfaça as duas desigualdades...

Notação θ

- Exemplo de constantes:

$$\left(c_1 n^2 \leq \frac{1}{2} n^2 - 3n \leq c_2 n^2 \right)$$ Dividindo por n^2 ...

$$= c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

Notação θ

- Exemplo de constantes:

$$\left(c_1 n^2 \le \frac{1}{2} n^2 - 3n \le c_2 n^2 \right)$$

Dividindo por n^2 ...

$$= c_1 \le \frac{1}{2} - \frac{3}{n} \le c_2 \quad \Longrightarrow \quad c_1 = \frac{1}{14}$$

$$c_2 = \frac{1}{2}$$

$$n_0 = 7$$

- Portanto, se existem tais constantes

$$\frac{n^2}{2} - 3n \in \Theta(n^2)$$

Notação θ

- Observação:

$$f(x) \in \Theta(g(x))$$

$$sse$$

$$f(x) \in O(g(x))$$

$$e$$

$$f(x) \in \Omega(g(x))$$

Notação *o (o minúsculo)*

Notação *o*

- O limite assintótico superior fornecido pela notação O (ó-zão) pode:

Notação *o*

- O limite assintótico superior fornecido pela notação O (ó-zão) pode:
  - Ser assintoticamente restrito;

Notação *o*

- O limite assintótico superior fornecido pela notação O (ó-zão) pode:
  - Ser assintoticamente restrito;
  - Não ser assintoticamente restrito;

Notação *o*

- O limite assintótico superior fornecido pela notação O (ó-zão) pode:
  - Ser assintoticamente restrito;
  - Não ser assintoticamente restrito;

- Exemplos:
  - Assintoticamente restrito: $2 \cdot n^2 \in O(n^2)$

  - Não assintoticamente restrito: $2n \in O(n^2)$

$$\log(n) \in O(c^n)$$

Notação *o*

- Todas as funções de O (ó-zão) que não definem um limite assintoticamente restrito pertencem a ‘’o” (ó-zinho)

Notação *o*

- Todas as funções de O (ó-zão) <u>que não definem</u> um limite assintoticamente restrito pertencem a ‘’o” (ó-zinho)

$$se\ \ f(n) \in O(g(n))\, e\ f(n) \notin \Omega(g(n))\, entao$$
$$f(n) \in o(g(n))$$

Notação *o*

- Todas as funções de O (ó-zão) que não definem um limite assintoticamente restrito pertencem a ‘’o” (ó-zinho)

$$se \;\; f(n) \in O(g(n)) \, e \, f(n) \notin \Omega(g(n)) \, entao$$

$$f(n) \in o(g(n))$$

$$2n \in o(n^2)$$

$$\log(n) \in o(n)$$

Notação *o*

$$o(g(n)) = \{f(n):\ \ \forall c > 0, \exists\, n_0 > 0$$
$$\mid 0 \le f(n) < c \cdot g(n) \ \ \forall \ \ n \ge n_0\}$$

- Comparativo com a notação O;

Não é <=, é somente <

$$f(n) \in O(g(n)), \text{o limite } 0 \le f(n) \le cg(n) \text{ se mantém válido}$$
$$\text{para alguma constante c} > 0$$
$$f(n) \in o(g(n)), \text{o limite } 0 \le f(n) < cg(n) \text{ é válido}$$
$$\text{para todas as constantes c} > 0$$

Notação *o*

- Facilitando o entendimento...

Se $f(n) \in o(g(n))$ então

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Notação *ω (omega minúsculo)*

Notação ω

- O limite assintótico inferior fornecido pela notação Ω (omega-zão) pode:
  - Ser assintoticamente restrito;
  - Não ser assintoticamente restrito;

- Exemplos:
  - Assintoticamente restrito: $$2 \cdot n^2 \in \Omega(n^2)$$

  - Não assintoticamente restrito: $$2 \cdot n^3 \in \Omega(n)$$

Notação ω

- Todas as funções de Ω (omegazão) que não definem um limite assintoticamente restrito pertencem a ω

$$se \ \ f(n) \notin O(g(n))\, e \ f(n) \in \Omega(g(n))\, entao$$
$$f(n) \in \omega(g(n))$$

$$2n^2 \in \omega(1)$$
$$2n \in \omega(\log(n))$$

Notação ω

$$\omega(g(n)) = \{f(n): \ \forall c > 0, \exists\, n_0 > 0$$

$$|\ 0 \leq c \cdot g(n) < f(n) \quad \forall \ \ n \geq n_0\}$$

Não é <=, é somente <

Notação ω

- Facilitando o entendimento...

$$\text{Se } f(n) \in \omega(g(n)) \text{ então}$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

Exercícios

Exercícios – V ou F

(a) $se\ f(n) \in \omega(g(n))\ entao\ f(n) \in \Omega(g(n))$

(b) $se\ f(n) \in \Omega(g(n))\ entao\ f(n) \in \omega(g(n))$

(c) $se\ f(n) \in o(g(n))\ entao\ f(n) \in O(g(n))$

(d) $se\ f(n) \in O(g(n))\ entao\ f(n) \in o(g(n))$

(e) $se\ f(n) \in \Theta(g(n))\ entao\ f(n) \in O(g(n))$

(f) $se\ f(n) \in \Theta(g(n))\ entao\ f(n) \in \Omega(g(n))$

(g) $se\ f(n) \in \Theta(g(n))\ entao\ f(n) \in o(g(n))$

(h) $se\ f(n) \in \Theta(g(n))\ entao\ f(n) \in \omega(g(n))$

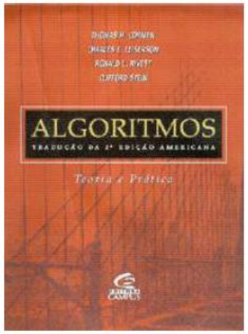
(i) $f(n) \in \omega(g(n))\ sse\ g(n) \in o(f(n))$

Leitura para próxima aula

- Livro: Algoritmos (Cormen)
  - 4 Recorrências;
    - 4.1 O método de substituição;
    - 4.2 O método de árvore de recursão
    - 4.3 O método mestre

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos – Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus.



- TAMASSIA, ROBERTO; GOODRICH, MICHAEL T. (2004). Projeto de Algoritmos - Fundamentos, Análise e Exemplos da Internet.